

API文档-云盾身份认证（实名二要素）

一. 产品说明

二. 产品功能

三. 产品接入说明

1. 入参说明

2. 入参获取

2.1. 查看appcode

2.2. 查看主账号ID

2.3. 创建verifyKey

3. 产品调用模式

四. 产品接入Demo

1. Java POM依赖:

2. Java代码Demo

3. 授权模式验签

1) 方式一：callBack回到接入的应用系统

2) 方式二：仅支持mini框的接入方式，不需要传入callBack参数

五. 出参说明

一. 产品说明

接入问题及获取无授权模式请加钉钉群：21727262

二. 产品功能

1. 功能说明：通过验证用户身份证号和姓名是否一致，来确定用户身份是否虚假，达到用户合规性要求。
2. 使用流程：
 - 步骤一：购买产品，见本页“立即购买”
 - 步骤二：产品接入，见本页“请求示例”

三. 产品接入说明

1. 入参说明

名称	类型	是否必选	说明
appcode	STRING	必选	调用产品的身份认证, 查看appcode值： https://market.console.aliyun.com/imageconsole/index.htm?#/bizlist
verifyKey	STRING	必选	调用产品的身份认证, 通过控制台创建verifyKey (必须由主账号创建) : https://yundunnext.console.aliyun.com/?p=saf#/config
_userId	STRING	必选	购买产品的主账号ID, 查看主账号ID： https://account.console.aliyun.com/#/secure
customerID	STRING	可选	用户账号ID, 不影响实时验证结果, 可用于问题排查和定位
identifyNum	STRING	可选	需验证用户的身份证
userName	STRING	可选	需验证用户的姓名

2. 入参获取

2.1. 查看appcode

- 购买产品之后才有appcode
- <https://market.console.aliyun.com/imageconsole/index.htm?#/bizlist>



The screenshot shows the AliCloud Market Console interface. On the left, there's a sidebar with categories like '订单列表', '需求管理', '定制方案管理', '退款管理', '发票管理', '优惠券管理', and '合同管理'. The main area is titled '已购买的服务' (Services Purchased). It lists a single item: '云盾身份认证(二要素)'. Below the item, it shows 'AppKey: 20...007' and 'AppSecret: 34nw...'. At the bottom of the item card, there are 'AppCode: cb2281e...' and '000ca11de7...' with '复制' (Copy) and '重置' (Reset) buttons. The entire 'AppCode' section is highlighted with a red box.

2.2. 查看主账号ID

- 需要主账号登录阿里云官网
- 地址: <https://account.console.aliyun.com/#/secure>

The screenshot shows the Alibaba Cloud account console at account.console.aliyun.com/#/secure. The left sidebar has tabs for Account Management, Security Settings, Basic Information, Real-name Authentication, Address Management, Student Authentication, Contact Person Management, Member Benefits, Member Points, and Cloud Agent. The main content area is titled 'Security Settings' and displays account information: 登录账号: 104863...com 修改 (您已通过实名认证) 第三方账号绑定 账号ID: 181...320252 注册时间: 2014年12月9日上午10:48:00. There is a '修改头像' (Change Avatar) button below the profile picture. Below the profile, it says 'Your current account security level' (您当前的账号安全程度) is 'Medium' (中), with a progress bar showing 50%. It also says 'Safety level: Medium' (安全级别: 中). A note says 'A strong password can make your account safer. It is recommended to change your password regularly, set a password containing letters, symbols, or numbers, and have a length of at least 6 digits.' (安全性高的密码可以使账号更安全。建议您定期更换密码，设置一个包含字母、符号或数字中至少两项且长度超过6位的密码。) A green checkmark indicates '已设置' (Set) and a link to '修改' (Modify).

2.3. 创建verifyKey

- VerifyKey必须由阿里云主账号创建
- 地址: <https://yundunnext.console.aliyun.com/?p=saf#/config>

1. 点击“新增VerifyKey”:

The screenshot shows the 'Cloud Protection - Risk Identification' section of the Yundun console. Under '接入统计' (Access Statistics), it shows '接入VerifyKey列表' (List of Access VerifyKeys). A red box highlights the '新增VerifyKey' (Add VerifyKey) button. Another red box highlights the '风险认证配置' (Risk Authentication Configuration) tab. The main table has columns for '操作端类型' (Operation End Type) and '域名白名单' (Domain White List), both currently empty. A note at the bottom says '没有数据' (No data).

2、填入域名（公司域名），选择操作端，然后点击确认

The screenshot shows the '接入设置' (Access Settings) page for adding a VerifyKey. The '接入域名' (Access Domain) field is labeled '请输入完整的域名' (Please enter the full domain name). The '操作端' (Operation End) dropdown is labeled '请选择' (Please select) and shows options 'MINI' and 'H5'. A large blue '确认' (Confirm) button is at the bottom. On the left, there is a collapsed '验证配置' (Verification Configuration) section.

3、页面返回VerifyKey信息



成功

添加成功,请记住如下系统分配的VerifyKey以及VerifySecret信息
短信签名已经提交审核, 审核成功之后方可生效

VerifyKey: [REDACTED]

VerifySecret: [REDACTED]... ...
[REDACTED]

复制

查看

3. 产品调用模式

身份证、姓名、手机号三要素涉及用户隐私数据，本产品提供两种用户授权方式：

1) 免授权模式

加钉钉群（21727262）找客服，咨询身份验证客户获取用户授权协议，加入自己业务的用户协议中，把用户协议地址给到客服确认后，客服配置免授权模式，接口直接返回认证结果信息。

2) 授权模式

按照请求示例接入，调通API后返回授权链接，在前端打开URL后用户点击授权后可以通过解签获取验证结果信息。具体验证流程见文档(入参获取和授权说明)。

四. 产品接入Demo

1. Java POM依赖：

```
1 <dependency>
2   <groupId>com.alibaba</groupId>
3   <artifactId>fastjson</artifactId>
4   <version>1.2.60</version>
5 </dependency>
6 <dependency>
7   <groupId>org.apache.httpcomponents</groupId>
8   <artifactId>httpclient</artifactId>
9   <version>4.2.1</version>
10 </dependency>
11 <dependency>
12   <groupId>org.apache.httpcomponents</groupId>
13   <artifactId>httpcore</artifactId>
14   <version>4.2.1</version>
```

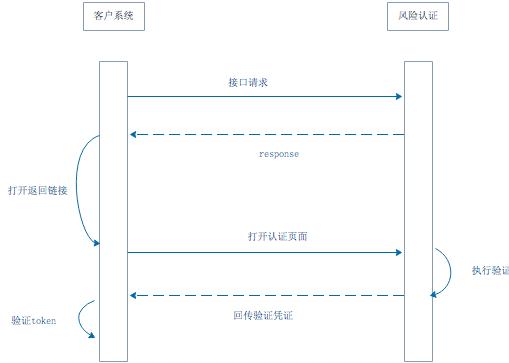
```
15 </dependency>
16 <dependency>
17     <groupId>commons-lang</groupId>
18     <artifactId>commons-lang</artifactId>
19     <version>2.6</version>
20 </dependency>
```

2. Java代码Demo

```
1 import java.util.HashMap;
2 import java.util.Map;
3
4 import org.apache.http.HttpResponse;
5 import org.apache.http.util.EntityUtils;
6
7 public class Safrv2metaIdName {
8
9     public static void main(String[] args) {
10         String host = "https://safrvcert.market.alicloudapi.com";
11         String path = "/safrv_2meta_id_name/";
12         String method = "GET";
13         // 查看链接: https://market.console.aliyun.com/imageconsole/in
14         // dex.htm?#/bizlist?_k=cgs1jr
15         String appcode = "自己的appcode";
16         Map<String, String> headers = new HashMap<String, String>();
17         //最后在header中的格式(中间是英文空格)为Authorization:APPCODE 8335
18         //9fd73fe94948385f570e3c139105
19         headers.put("Authorization", "APPCODE " + appcode);
20         Map<String, String> querys = new HashMap<String, String>();
21         querys.put("__userId", "主账号ID");
22         querys.put("customerID", "12345678");
23         querys.put("identifyNum", "身份证号");
24         querys.put("userName", "姓名");
25         //创建和查询链接: https://yundunnext.console.aliyun.com/?p=saf
#config
26         querys.put("verifyKey", "自己的verifyKey");
27
28 }
```

```
26     try {
27         /**
28          * 重要提示如下：
29          * HttpUtils请从
30          * https://github.com/aliyun/api-gateway-demo-sign-java/
31          * blob/master/src/main/java/com/aliyun/api/gateway/demo/util/HttpUtil
32          * s.java
33         */
34         HttpResponse response = HttpUtils.doGet(host, path, meth
35         od, headers, querys);
36         System.out.println(response.toString());
37
38         //查看云市场报错信息
39         //for (Header header : response.getAllHeaders()) {
40         //    if(header.getName().contains("X-Ca-Error-Messag
41         //e")) {
42             //        System.out.println("Error message: " + header.
43             //getValue());
44         //    }
45         //}
46     }
47 }
48 }
```

3. 授权模式验签



用户在认证页面，身份验证成功之后，验证页面会把验证结果token回传给接入应用，这个验证结果token是标识当次身份验证是否有效的关键，防止恶意篡改，接入应用必须要对这个token进行签名验证。

- 身份验证页面回传token给接入应用的两种方式：

1) 方式一：callback回到接入的应用系统

比如在无线端接入身份验证的H5页面时，身份验证成功会跳回到接入应用的业务页面，并将token和签名信息放在url中带给接入应用。

- 获取token及签名的Demo：

```

1 String ivSign = request.getParameter("ivSign");
2 //进行验签操作...

```

2) 方式二：仅支持mini框的接入方式，不需要传入callback参数

通过javascript的postMessage机制发送消息告诉接入应用的业务页面。比如在PC上通过MINI弹窗来接入身份验证的页面时，身份验证成功会将成功的token和签名postMessage会接入应用的业务页面。

- 获取token及签名的Demo：

```

1 window.addEventListener("message", function( event ) {
2     var json = JSON.parse(decodeURIComponent(event.data));
3     if('type' in json && json.type=='iframevalid'){
4         var ivSign = json.ivSign;
5         //进行验签操作...
6     }
7 } );

```

对验证成功token进行验签，（通过前面申请的VerifySecret进行验签）

- 示例Java Demo:

```

1 /**
2  * 参数均是从请求中获取
3  * @param ivSign    验证成功之后的回传参数sign
4  * @param verifySecret 在接入时分配的verifySecret
5 */
6     public static void verifySign(String ivSign, String verifySecret)
7     {
8         try{
9             byte[] encryptedData = java.util.Base64.getDecoder().dec
ode(ivSign.getBytes("utf-8"));
10            byte[] keyData = java.util.Base64.getDecoder().decode(ve
rifySecret.getBytes());
11            PKCS8EncodedKeySpec keySpec = new PKCS8EncodedKeySpec(ke
yData);
12            KeyFactory keyFactory = KeyFactory.getInstance("RSA");
13            RSAPrivateKey privateKey = (RSAPrivateKey) keyFactory.ge
neratePrivate(keySpec);
14            Cipher cipher = Cipher.getInstance("RSA");
15            cipher.init(Cipher.DECRYPT_MODE, privateKey);
16            String content = new String(doBlock(encryptedData, priva
teKey.getModulus().bitLength()/8, cipher), "utf-8");
17            Map<String, String> map = JSON.parseObject(content,Map.cl
ass);
18            //校验通过
19            if(map!=null && map.containsKey("success")){
20                boolean valid = false;
21                String ivTs = map.get("signTs");
22                if(ivTs!=null && !"".equals(ivTs)){
23                    Long expiredTime = Long.parseLong(ivTs)+300;
24                    if(expiredTime>System.currentTimeMillis()/1000){
25                        valid = true;
26                    }
27                }
28                //签名未过期
29                if(valid){
30                    if(Boolean.valueOf(map.get("success"))){ //认证通过, 继续业务操作
31                        String customerId = map.get("customerId");

```

```
32             }else{
33                 //认证不通过
34             }
35         }
36     }else{
37         //签名无效 阻断
38         return;
39     }
40 }catch (Exception e){
41
42 }
43 }
44
45     private static byte[] dolock(byte[] encryptedData, int blockSize
46 , Cipher cipher) throws IllegalBlockSizeException, BadPaddingException {
47
48     int inputLen = encryptedData.length;
49     ByteArrayOutputStream out = new ByteArrayOutputStream();
50     int offSet = 0;
51     byte[] cache;
52     int i = 0;
53     // 对数据分段解密
54     while (inputLen - offSet > 0) {
55         if (inputLen - offSet > blockSize) {
56             cache = cipher.doFinal(encryptedData, offSet, blockSize);
57         } else {
58             cache = cipher.doFinal(encryptedData, offSet, inputLen - offSet);
59         }
60         out.write(cache, 0, cache.length);
61         i++;
62         offSet = i * blockSize;
63     }
64
65     byte[] decryptedData = out.toByteArray();
66     IOUtils.closeQuietly(out);
67     return decryptedData;
68 }
```

- 示例PHP Demo:

```

1 function verifySign($ivSign,$verifySecret) {
2     $content="";
3     $verifySecret = chunk_split($verifySecret,64,"\\n");
4     $verifySecret = "-----BEGIN RSA PRIVATE KEY-----\\n".$verifyS
ecret."-----END RSA PRIVATE KEY-----\\n";
5     $pri_key = openssl_get_privatekey($verifySecret);
6     $key_len = openssl_pkey_get_details($pri_key)['bits'];
7     $part_len = $key_len / 8;
8     $encrypted = base64_decode(rawurldecode($ivSign));
9     $parts = str_split($encrypted, $part_len);
10    foreach ($parts as $part) {
11        $decrypted_temp = '';
12        openssl_private_decrypt($part, $decrypted_temp,$pri_ke
y);
13        $content .= $decrypted_temp;
14    }
15    $content = iconv("UTF-8",iconv_get_encoding("internal_encodi
ng"),$content);
16    $json = json_decode($content);
17    if(intval($json->{'signTs'})+300>time()){
18        //签名未过期, 继续业务操作
19        if($json->{'success'}=='true'){
20            //标识成功了
21        }else{
22            //标识失败
23        }
24    }else{
25        echo "expired";
26    }
27 }
```

五. 出参说明

- 免授权模式返回值示例

```

1 {
2     "code": 200,
3     "value": {
4         "bizCode": 0,
5         "message": "success"
6     },
7     "message": "success"
8 }

```

- 授权模式返回值示例

```

1 {
2     "code": 200,
3     "value": {
4         "verifyUrl": "https://yuniv.aliyuncs.com/iv/remote/h5/reques
t.htm?havana_iv_token=AQgAENgEIg1oYXZhbmFfYXBwX2l2MgEB0Pb97pPKLEABShB
VBpCDElX06cLLSyS6gnySaQyCdDcgxpqTC1u0d1AdnEfX-A"
5     },
6     "message": "success"
7 }

```

- 返回值说明

名称	类型	是否必选	说明
code	STRING	必选	调用码, 具体见 【调用码说明】
value.verifyUrl	STRING	可选	用户验证链接, 授权流程见下文 【授权说明】
value.bizCode	STRING	可选	业务码, 具体见 【业务码说明】
message	STRING	必选	返回信息说明

- 调用码code说明

code值	说明
200	调用成功
402	调用错误
403	无权限调用
500	内部服务错误

501

无此记录

- 调用码code=200时，业务码bizCode说明

bizCode值	说明
0	验证通过
13066	验证不一致